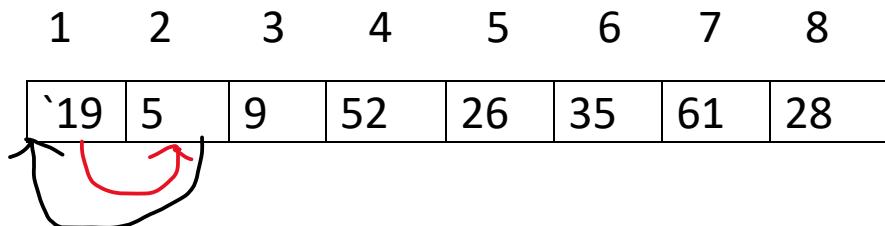
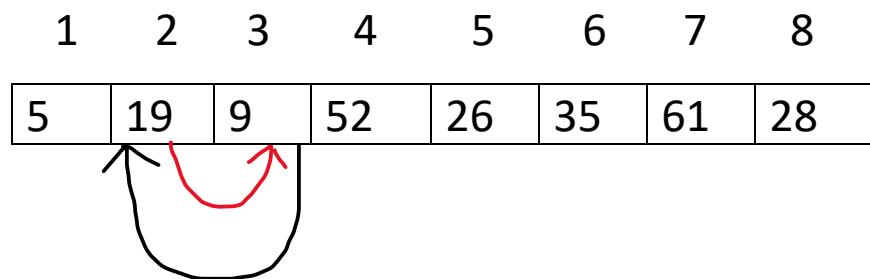


Q1] A) Illustrate sorting via insertion sort on the array  
A[19,5,9,52,26,35,61,28]

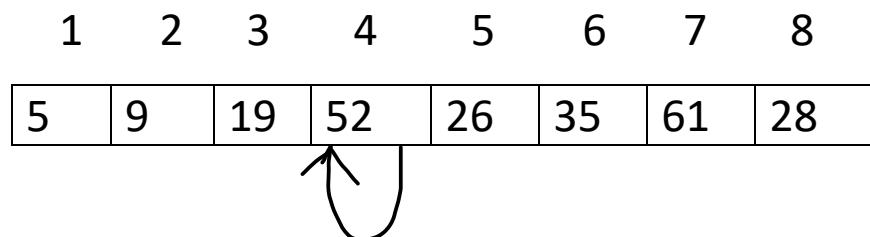
i.      J=2



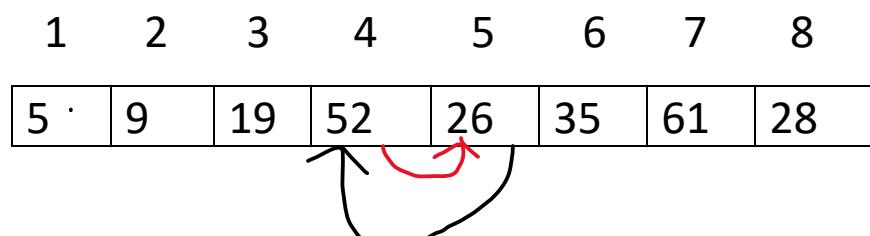
ii.     J=3



iii.    J=4

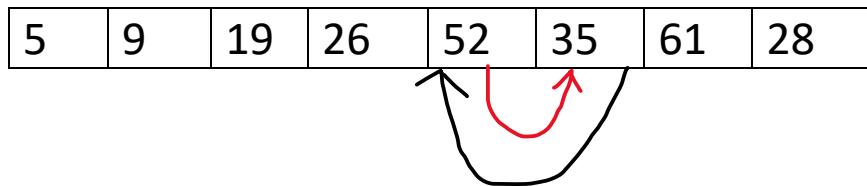


iv.     J=5



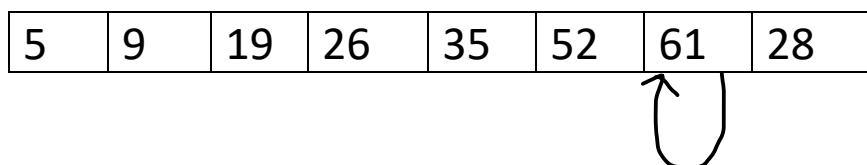
v.      J=6

1    2    3    4    5    6    7    8



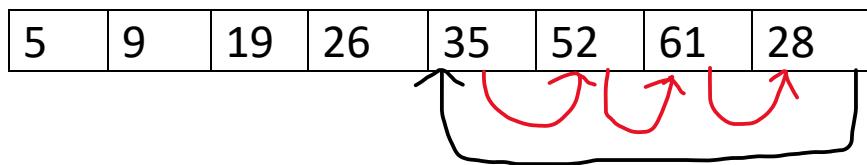
vi.      J=7

1    2    3    4    5    6    7    8



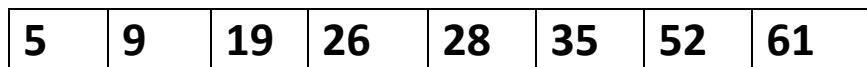
vii.      J=8

1    2    3    4    5    6    7    8



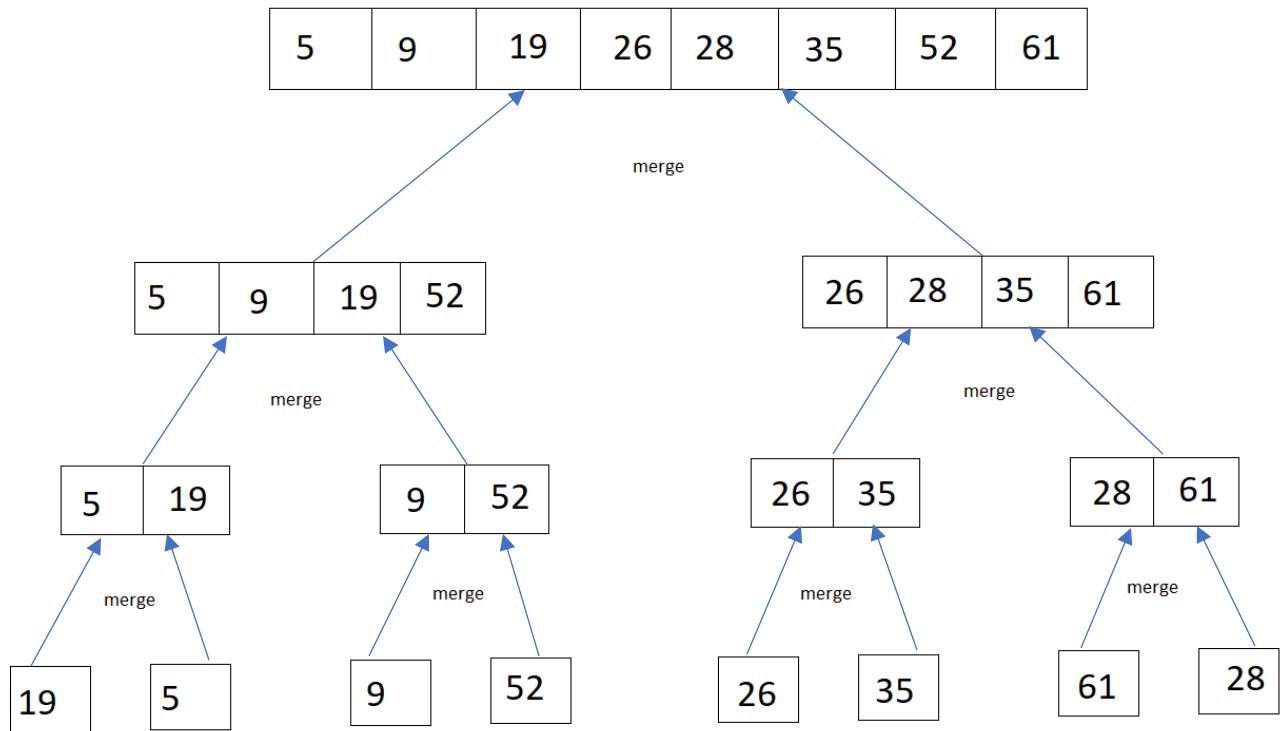
viii.      Final Sorted Array

1    2    3    4    5    6    7    8



Q1. B]

Illustrate sorting via merge sort on the array  
A[19,5,9,52,26,35,61,28]

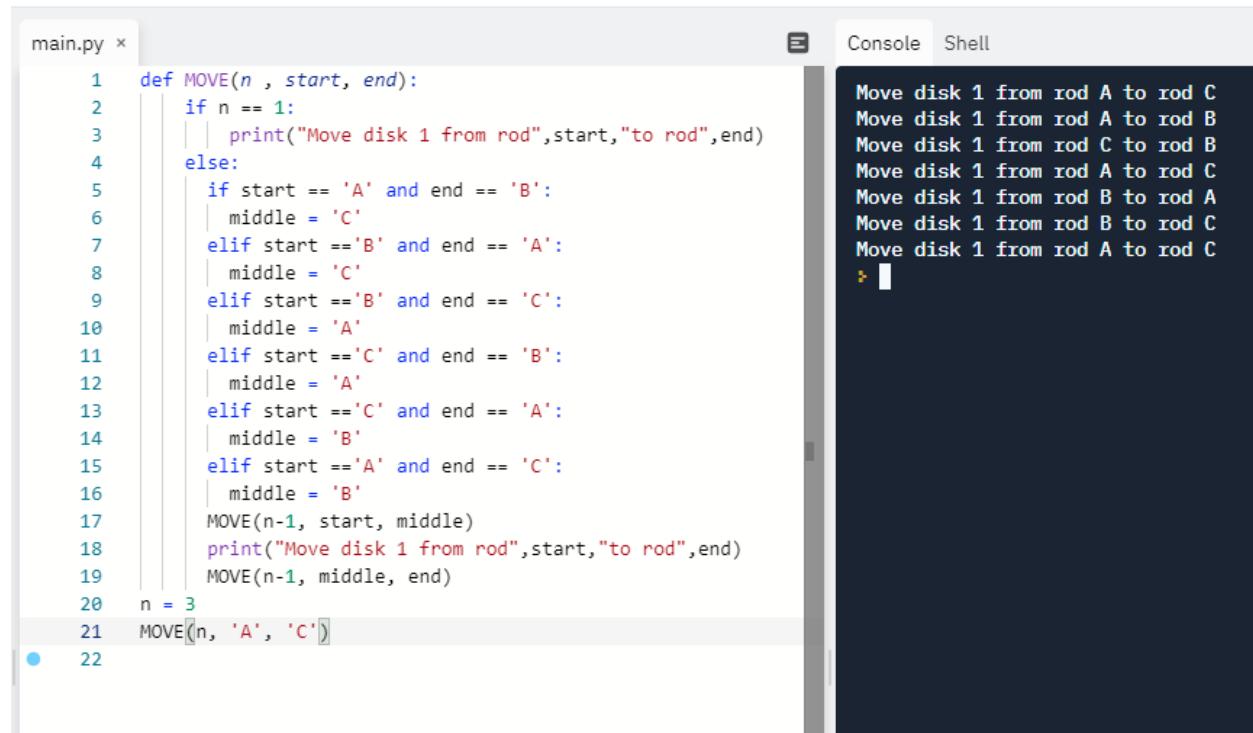


## Q4] Tower of Hanoi Pseudo Code

MOVE(n, start,end):

```
If n == 1 :  
    print("Move the top disk from rod", start, "to rod", end)  
  
else:  
    middle = third rod ..... // calculate from start and end rod  
    MOVE(n-1, start, middle)  
    Print("Move the top disk from rod", start, "to rod", end)  
    MOVE(n-1, middle, end)
```

Following is a screenshot of the code implemented from the above pseudo code for reference:



The screenshot shows a code editor window titled "main.py" and a terminal window titled "Console". The code in main.py implements the Tower of Hanoi algorithm using a recursive MOVE function. The terminal shows the output of the algorithm for n=3, which involves moving disks between three rods (A, B, and C).

```
main.py x  
1  def MOVE(n , start, end):  
2      if n == 1:  
3          print("Move disk 1 from rod",start,"to rod",end)  
4      else:  
5          if start == 'A' and end == 'B':  
6              middle = 'C'  
7          elif start =='B' and end == 'A':  
8              middle = 'C'  
9          elif start =='B' and end == 'C':  
10             middle = 'A'  
11         elif start =='C' and end == 'B':  
12             middle = 'A'  
13         elif start =='C' and end == 'A':  
14             middle = 'B'  
15         elif start =='A' and end == 'C':  
16             middle = 'B'  
17         MOVE(n-1, start, middle)  
18         print("Move disk 1 from rod",start,"to rod",end)  
19         MOVE(n-1, middle, end)  
20     n = 3  
21     MOVE(n, 'A', 'C')  
22
```

Console Output:

```
Move disk 1 from rod A to rod C  
Move disk 1 from rod A to rod B  
Move disk 1 from rod C to rod B  
Move disk 1 from rod A to rod C  
Move disk 1 from rod B to rod A  
Move disk 1 from rod B to rod C  
Move disk 1 from rod A to rod C  
> |
```